Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00279-5

Original Article

A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach

Ali Al-Bayaty [1], Shanyan Chen [2], Ali Al-Shuwaili [3], Abbas AlZubaidi [4], Marek Perkowski [1]

1. *Department of Electrical and Computer Engineering, Portland State University, Portland, USA*
2. *College of Information Engineering, Capital Normal University, Beijing, CHN*
3. *College of Engineering, University of Information Technology and Communications, Baghdad, IRQ*
4. *College of Healthcare Technologies, American University of Iraq-Baghdad, Baghdad, IRQ*

Received: January 24, 2026 | Review began: February 03, 2026 | Review ended: August 15, 2026 | Published: September 15, 2026

Abstract

The conventional design technique of any n-bit quantum gate is mainly achieved using the multiplication of unitary matrices for $n - m$ control qubits and m target qubits, where $n \geq 2$ and $1 \leq m \leq n - 1$. These matrices represent quantum rotations by an n-bit quantum gate. For a quantum designer, such a conventional technique requires extensive computational time and effort, which may generate an n-bit quantum gate with too high quantum cost. In general, the Bloch sphere is only utilized as a visualization tool to verify the conventional design correctness for quantum rotations by a quantum gate. In contrast, this paper introduces a new concept of using the Bloch sphere as a "geometrical design tool" to build cost-effective n-bit quantum gates with lower quantum costs. This concept is termed the "Bloch sphere approach (BSA)." In BSA, a cost-effective n-bit quantum gate is built without using any multiplication of unitary matrices. Instead, the quantum rotations for such a gate are visually selected using the geometrical planar intersections of the Bloch sphere. The BSA can efficiently map m targets among $n - m$ controls for an n-bit quantum gate to satisfy the limited layout connectivity for the physical neighboring qubits of a superconducting quantum computer. Experimentally, n-bit quantum gates built using the BSA always have lower quantum costs than those for such gates built using the conventional quantum design techniques, by utilizing the same transpilation settings and the restricted layout connectivity of a real IBM superconducting quantum computer, where $2 \leq n \leq 5$ qubits.

Categories: Quantum Algorithms, Quantum Computing, HPC Applications

Keywords: quantum rotations, bloch sphere approach, quantum gates, quantum circuits synthesis, geometrical design tool, weighted transpilation quantum cost

Introduction

In quantum computing, an *n*-bit quantum gate of *m* target qubits and $\boldsymbol{n - m}$ control qubits can be conventionally designed through the multiplication approach of its complete set of unitary matrices [1-4], where $\boldsymbol{n \geq 2}$ and $\boldsymbol{1 \leq m \leq n - 1}$. Every unitary matrix represents a single quantum rotation for such an *n*-bit quantum gate. In general, some *n*-bit quantum gates are fundamentally used to construct other *n*-bit quantum gates. For instance, the *n*-bit Toffoli gate [1-4] is mainly used to construct the *n*-bit Fredkin [5-8], *n*-bit Miller [7-10], and *n*-bit Peres and *n*-bit inverse Peres [7,8,10,11] gates. Subsequently, an *n*-bit quantum gate can be completely redesigned using the phase polynomial approach [12,13]. Besides, the correctness of this phase-based approach can also be algebraically verified by multiplying unitary matrices.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Theoretically, on the one hand, Barenco et al. [1] discussed the construction of the 3-bit Toffoli gate using the multiplication of unitary matrices. On the other hand, Shende and Markov [12] also constructed the 3-bit Toffoli gate using the phase polynomial approach. Both approaches aimed to design the quantum counterpart of a classical Boolean AND gate of two inputs, i.e., two control qubits, and one output, i.e., one target qubit. In addition, the correctness of these two conventional approaches can be visually verified using the geometrical quantum rotations on the Bloch sphere. Hence, the Bloch sphere is mainly utilized in quantum computing as the "geometrical visualization tool" for only verifying the final conventional design of any *n*-bit quantum gate using a distinct qubit, e.g., the Bloch sphere can verify the design correctness of a 3-bit Toffoli gate by visualizing the quantum-state transition of its output qubit for all permutative states of the two control qubits.

Practically, we noticed that these two conventional approaches do not generate efficient *n*-bit quantum gates in the context of certain defined higher quantum costs. In this paper, the quantum cost is the total number of generated native (basis) gates for a transpiled (mapped) quantum circuit into the layout (architecture) of a real quantum computer, i.e., a quantum processing unit (QPU). The following three observations lead to proposing this research for minimizing quantum costs:

1. Every QPU has a limited connectivity layout for a number of physical neighboring qubits, e.g., the ibm_torino QPU of 133 qubits has the heavy-hex layout of up to four connected physical neighboring qubits [14,15]. Notice that such a limited connectivity layout is not taken into account during the design stage, i.e., the conventional design for an *n*-bit quantum gate is not a layout-aware approach.

2. Every QPU has a defined supported set of native gates that are generated in the final transpiled quantum circuits, e.g., the ibm_torino QPU has the native gates of identity (I), Pauli-X (X), rotational X (RX), half-rotational X ($\sqrt{\mathbf{X}}$), rotational Pauli-Z (RZ), rotational ZZ (RZZ), and controlled-Z (CZ), where their quantum rotations are defined in $\frac{\pi}{k}$ radians and $k \in \mathbb{R}$. Notice that, for current IBM QPUs, the I, X, RX, $\sqrt{\mathbf{X}}$, and RZ are single-qubit native gates, and the RZZ and CZ are two-qubit native gates.

3. An *n*-bit quantum gate is, generally, transpiled into a QPU with a large set of native gates due to the non-native gates decomposition. That means the final transpiled quantum circuit always has a higher quantum cost.

Please observe that the limited connectivity layout of a QPU could add additional SWAP gates [16,17] to connect the physical non-neighboring qubits for a transpiled quantum circuit, which increases its quantum cost as well. A higher quantum cost could decrease the fidelity of a transpiled quantum circuit, by increasing the physical qubits decoherence that may introduce errors in the final measured results [18-20].

Based on the aforementioned three observations, the goal of this paper is to introduce a new geometrical design approach for building cost-effective *n*-bit quantum gates in the context of lower quantum costs and layout-aware efficient mapping into a specific QPU. This new geometrical design approach does not require any multiplication of unitary matrices and phase polynomial generation techniques. Instead, our approach mainly utilizes the Bloch sphere itself as a "geometrical design tool" for visually building cost-effective *n*-bit quantum gates. This new approach is termed the "Bloch sphere approach (BSA)". Such that, in the BSA, the quantum rotations for the to-be-built *n*-bit quantum gate are visually selected using the geometrical planar intersections with the Bloch sphere. These geometrical planar intersections are the functional points in the XY-plane (equator) of the Bloch sphere. In our research, we classify these functional points into "octants," "quadrants," and "semicircles," due to the fact that all quantum rotations of the supported native gates are performed on this XY-plane for a specific QPU with the defined set of native gates and the layout geometry. For instance, the following native gates of the ibm_torino QPU: (i) RZ($+\frac{\pi}{4}$) as "T" and RZ($-\frac{\pi}{4}$) as "$\mathbf{T}^{\dagger}$" gates perform their quantum rotations on the octants, (ii) RZ($+\frac{\pi}{2}$) as "S" and RZ($-\frac{\pi}{2}$) as "$\mathbf{S}^{\dagger}$" gates perform their quantum rotations on the quadrants, and (iii) RZ($+\pi$) as "Z" gates perform their quantum rotations on the semicircles of the XY-plane. With respect to the layout geometry, the BSA effectively enhances the aforementioned three observations as follows.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

1. Mapping m target qubits in the middle of $n-m$ control qubits of an *n*-bit gate without adding SWAP gates to connect the physical non-neighboring qubits, to overcome the limited connectivity layout for a specific QPU.

2. The generated *n*-bit gate is initially built from the supported native gates of a specific QPU.

3. The generated *n*-bit gate is transpiled into a specific QPU with the same initial number of supported native gates, since no further gate decomposition is required. That means its final transpiled circuit always has a lower quantum cost.

Subsequently, using the BSA, various cost-effective *n*-bit quantum gates and libraries are designed for IBM quantum computers using the symmetrical circuit structure [1,7,8,21,22], Clifford+T gates [3,8,22,23], and IBM native gates. Our two designed cost-effective *n*-bit quantum libraries are the generic architecture of layout-aware *n*-bit (GALA-n) operators [7] and Clifford+T-based architecture of layout-aware *n*-bit (CALA-n) operators [8], which both consist of (i) *n*-bit Toffoli gates, (ii) *n*-bit Boolean (AND, NAND, OR, NOR, Implication, and Inhibition) gates [7,8,24], (iii) *n*-bit controlled-$\sqrt{\mathbf{X}}$ (CV) and *n*-bit controlled-$\sqrt{\mathbf{X}}^{\dagger}$ ($\mathbf{CV}^{\dagger}$) gates [1,7,8], (iv) *n*-bit Fredkin gates, (v) *n*-bit Miller gates, and (vi) Boolean-phase SWAP "p-SWAP" gate [22], where $2 \leq n \leq 5$ qubits and $-\pi \leq p \leq +\pi$ radians. Notably, both GALA-n and CALA-n quantum libraries have become part of the IBM Qiskit Ecosystem [25].

In this research, we also introduce a new generic metric for efficiently calculating the quantum cost for the final transpiled quantum circuits into a specific QPU. This is an addition to several costs introduced in the past [9,26]. We term our metric the "weighted transpilation quantum cost (WTQC)", which is an improved variant of our previously proposed TQC [7,27]. The WTQC is the weighted sum of four components: (i) the single-qubit native gates, (ii) the two-qubit native gates, (iii) the SWAP gates, and (iv) the depth for the final transpiled quantum circuit, where each component has its own weight "W_i" for importance ($W_i \geq 0$). Notably, the WTQC is a technology-dependent cost metric, while Maslov cost [26] is a technology-independent cost metric that does not calculate the decomposed non-native gates after transpilation, e.g., the decomposed 3-bit Toffoli gates. Notice that Maslov cost is good for fast approximate cost evaluation, while the WTQC is more appropriate for evaluating a transpiled circuit for a specific QPU. Experimentally, after transpilation with an IBM QPU, we prove that different *n*-bit quantum gates and operators built using the BSA always have a lower WTQC than those gates built using the conventional design approaches.

Materials And Methods

The Bloch sphere

The Bloch sphere is a geometrical sphere in a three-dimensional space with three X, Y, and Z spatial axes, which represents the quantum states of a qubit. When a series of quantum gates is applied to a qubit, the Bloch sphere visualizes such quantum operations in Hilbert space [3,4]. Therefore, the Bloch sphere is commonly utilized in quantum computing as a geometrical visualization (and verification) tool. Figure *1* depicts six distinct base states of a qubit, and each visualizes an intersection of one of the three spatial axes with the Bloch sphere.

An *n*-bit quantum gate having X (as a Pauli-X), Y (as a Pauli-Y), or Z (as a Pauli-Z) in its notation rotates the states of a qubit around the X, Y, or Z axis within the Bloch sphere, respectively, where $n \geq 1$. Such a quantum rotation around a spatial axis relies on a defined rotational angle ($\pm\theta$) expressed in radians, as shown in Figure *1*, where $+\theta$ represents a counterclockwise rotation and $-\theta$ represents a clockwise rotation for $-\pi \leq \theta \leq +\pi$. For instance, the following Z gates rotate the states of a qubit by the Z-axis for the defined θ angles:

1. The Z gate performs the quantum rotation of $+\pi$ radians.

2. The $\sqrt{\mathbf{Z}}$ gate performs the quantum rotation of $+\frac{\pi}{2}$ radians. The $\sqrt{\mathbf{Z}}$ gate is the so-called "S" gate [1,3,4].

3. The $\sqrt[4]{Z}$ gate performs the quantum rotation of gate performs the quantum rotation of $+\frac{\pi}{4}$ radians. The $\sqrt[4]{Z}$ gate performs the quantum rotation of gate is the so-called "T" gate [3,4,13].

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Notice that all *n*-bit quantum gates are unitary gates, and a few of them are non-Hermitian gates [1,3,4], i.e., their quantum operations are not equal to their own inverses, as the aforementioned S and T gates. In addition, some rotational gates alter the global phase for certain states of a qubit. Thus, the choice of rotational gates can be a critical factor in circuit design. For instance, the Z gate is not the same as the IBM native RZ($+\pi$) gate, due to the global phase difference of $-i$ between the two gates. Such that, $R_Z(+\pi) = e^{-i\frac{\pi}{2}\mathrm{Z}} = \begin{bmatrix} e^{-i\frac{\pi}{2}} & 0 \\ 0 & e^{+i\frac{\pi}{2}} \end{bmatrix} = -i\mathbf{Z}$.

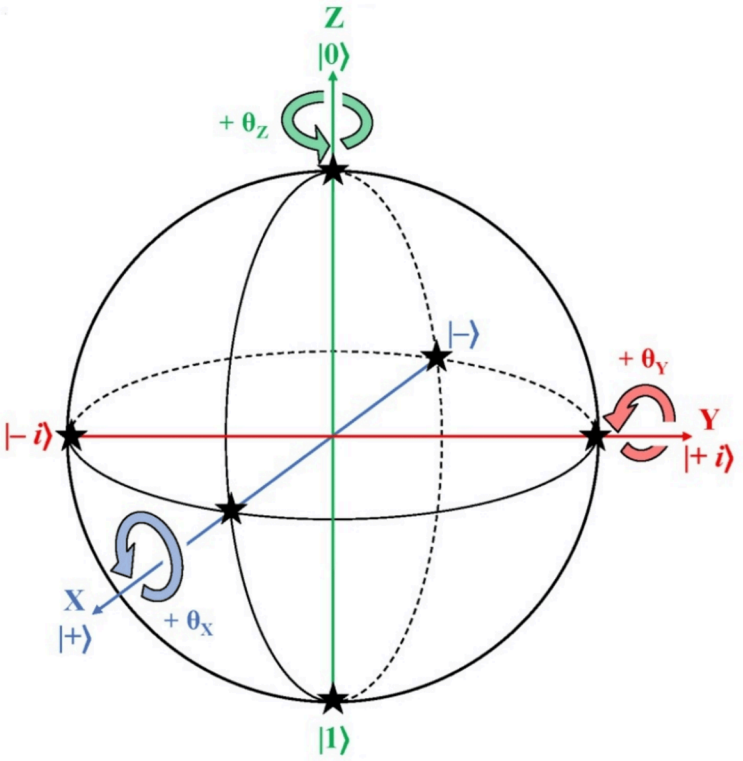

FIGURE 1: The Bloch sphere is a three-dimensional space with three spatial axes (X, Y, and Z) for illustrating six distinct base states (the stars) of a qubit.

The Bloch sphere is a three-dimensional space with three spatial axes (X, Y, and Z) for illustrating six distinct base states (the stars) of a qubit: (i) the $|\mathbf{0}\rangle$ and $|\mathbf{1}\rangle$ states intersecting the Z-axis with the Bloch sphere, (ii) the $|+\rangle$ and $|-\rangle$ states intersecting the X-axis with the Bloch sphere, and (iii) the $|+i\rangle$ and $|-i\rangle$ states intersecting the Y-axis with the Bloch sphere.

Clifford+T gates and symmetrical circuit structures

The Clifford+T gates (as a Clifford group + T gates) [3,8,22,23] are a set of (i) single-qubit gates, which are I, X, Y, Z, Hadamard (H), $\sqrt{\mathbf{X}}$, $\sqrt{\mathbf{X}}^{\dagger}$, S, $\mathbf{S}^{\dagger}$, T, and $\mathbf{T}^{\dagger}$, and (ii) two-qubit gates, which are controlled-X (CX, CNOT, or Feynman), controlled-Y (CY), CZ, and SWAP. The Clifford group plays a significant role in quantum computing, because they: (i) invert the phase of a Pauli gate, and (ii) construct a Pauli gate from other Pauli gates, as expressed in Eq. (1) and stated in Table *1*, where C is a Clifford group gate, $\mathbf{P}_{\sigma}$ is a Pauli gate, $\mathbf{P}_{\Sigma}$ is the resultant Pauli gate, the symbol (†) is the conjugate transpose (adjoint) operator, and the symbol (·) is the matrix multiplication operator [1,3,4,8,22,23].

$$\mathbf{C} \cdot \mathbf{P}_{\sigma} \cdot \mathbf{C}^{\dagger} = \mathbf{P}_{\Sigma} \tag{1}$$

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

C	·	P_σ	·	$C^\dagger$	=	P_Σ
H	·	Z	·	H	=	X
S	·	X	·	$S^\dagger$	=	Y
H	·	X	·	H	=	Z
Z	·	X	·	Z	=	– X
Z	·	Y	·	Z	=	– Y
X	·	Z	·	X	=	– Z

TABLE 1: Transformation examples of Clifford group for inverting the phases of Pauli gates and constructing other Pauli gates.

For different quantum computers, the Clifford+T gates (CTG) are mostly supported as native gates. For instance, based on the native gates of ibm_torino QPU, the Clifford+T gates are limited to the set {X, Z, H, $\sqrt{\mathbf{X}}$, S, $\mathbf{S}^\dagger$, T, $\mathbf{T}^\dagger$, CZ}, which is denoted here by $\mathbf{CTG_0}$. On the one hand, the gates of $\mathbf{CTG_0}$ rotating around the X-axis of the Bloch sphere are the X and $\sqrt{\mathbf{X}}$ gates. On the other hand, the gates of $\mathbf{CTG_0}$ rotating around the Z-axis of the Bloch sphere are the Z, S, $\mathbf{S}^\dagger$, T, $\mathbf{T}^\dagger$, and CZ gates. The H gate of $\mathbf{CTG_0}$ is primarily used in quantum computing to create superposition states, by transforming the state of a qubit from the Z-axis ($|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$) to the XY-plane ($|+\rangle \boldsymbol{or} |-\rangle$), respectively, and vice versa. In general, the H, $\sqrt{\mathbf{X}}$, and $\sqrt{\mathbf{X}}^\dagger$ can be utilized as superposition gates. However, different analyses should be taken into account when utilizing $\sqrt{\mathbf{X}}$ and $\sqrt{\mathbf{X}}^\dagger$ as superposition gates to build cost-effective *n*-bit gates for various quantum computers [28].

Notably, for current IBM QPUs, (i) the X and $\sqrt{\mathbf{X}}$ gates are single-qubit native gates, (ii) the H gate is a single-qubit non-native gate that can be decomposed into the set {S, $\sqrt{\mathbf{X}}$, S}, (iii) the Z, S, $\mathbf{S}^\dagger$, T, and $\mathbf{T}^\dagger$ gates can be utilized as single-qubit native RZ($+\boldsymbol{\pi}$), RZ($+\frac{\boldsymbol{\pi}}{\mathbf{2}}$), RZ($-\frac{\boldsymbol{\pi}}{\mathbf{2}}$), RZ($+\frac{\boldsymbol{\pi}}{\mathbf{4}}$), and RZ($-\frac{\boldsymbol{\pi}}{\mathbf{4}}$) gates, respectively, (iv) the CZ gate is a two-qubit native gate, and (v) the CNOT gate is a two-qubit non-native gate that can be decomposed into the set {H, CZ, H}, where the H gates are placed for the target qubit. Therefore, these non-decomposed and decomposed Clifford+T gates are IBM native gates.

For $\boldsymbol{n \geq 2}$, an *n*-bit quantum gate has $\boldsymbol{n-1}$ controls (input) qubits and one target (output) qubit, except for the SWAP, Fredkin, or a specific-purpose gate. All $\boldsymbol{n-1}$ controls are connected to one target. For symmetrical circuit structures, by routing (re-mapping) the target among $\boldsymbol{n-1}$ controls, an *n*-bit quantum gate can then be cost-effectively fitted into the limited connectivity layout of $\boldsymbol{n}$ physical neighboring qubits for any quantum computer. Hence, for such an *n*-bit quantum gate, SWAP gates are never added to connect the physical non-neighboring qubits (specifically for $\boldsymbol{3 \leq n \leq 4}$, since all superconducting quantum computers support the square grids or heavy-hex layouts of three to four physical non-neighboring qubits), and the final quantum cost for its transpiled quantum circuit is significantly decreased.

Figure *2* shows arbitrary *n*-bit symmetrical quantum gates for $\boldsymbol{2 \leq n \leq 4}$, where $\mathbf{SP_1}$ and $\mathbf{SP_2}$ are the superposition gates of H, $\sqrt{\mathbf{X}}$, and $\sqrt{\mathbf{X}}^\dagger$, $\boldsymbol{\theta}$'s are Clifford+T gates {Z, S, $\mathbf{S}^\dagger$, T, $\mathbf{T}^\dagger$} and the angles of IBM native RZ($\pm\boldsymbol{\theta}$) gates, and all $\boldsymbol{\theta}$'s may have similar values.

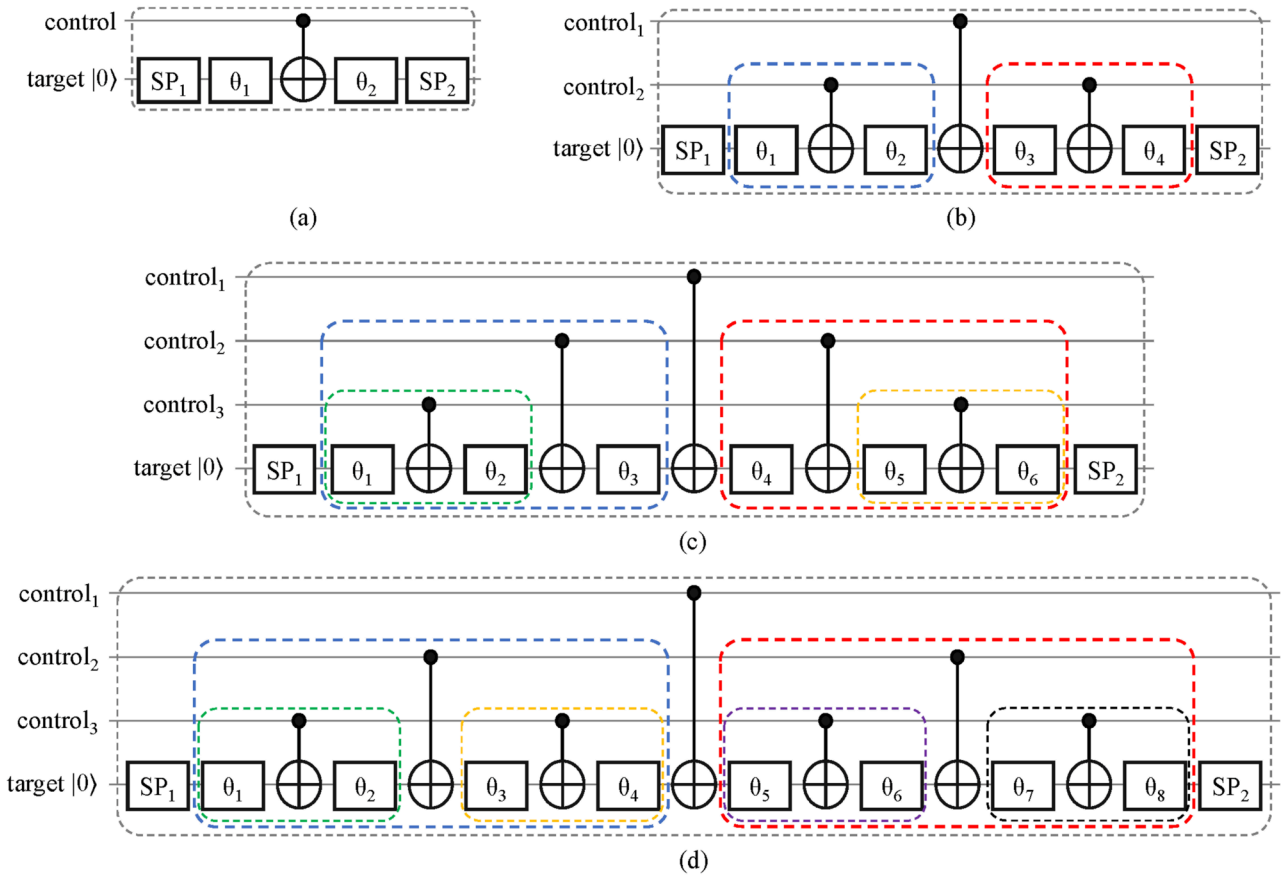

FIGURE 2: Arbitrary n-bit symmetrical quantum gates.

(a) A 2-bit symmetrical gate, (b) a 3-bit symmetrical gate, (c) a 4-bit symmetrical gate, and (d) a 4-bit symmetrical gate, where $\mathbf{2} \leq \boldsymbol{n} \leq \mathbf{4}$ and the colored rectangles denote symmetries [27,29].

The Bloch sphere approach

Geometrically, the coordinate and parallel planes are two-dimensional planes perpendicular to the individual three-coordinate spatial axes within the Bloch sphere. Figure *3* illustrates different coordinate and parallel planes, as the XY-plane, XZ-plane, YZ-plane, and $\boldsymbol{m}$ parallel planes ($\boldsymbol{P}$) perpendicular to various of the three coordinate axes with the Bloch sphere, where $\boldsymbol{m} \geq \mathbf{1}$ and the black dots denote the segments of "semicircles," "quadrants," and "octants" for the resulting circular intersections with the Bloch sphere.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

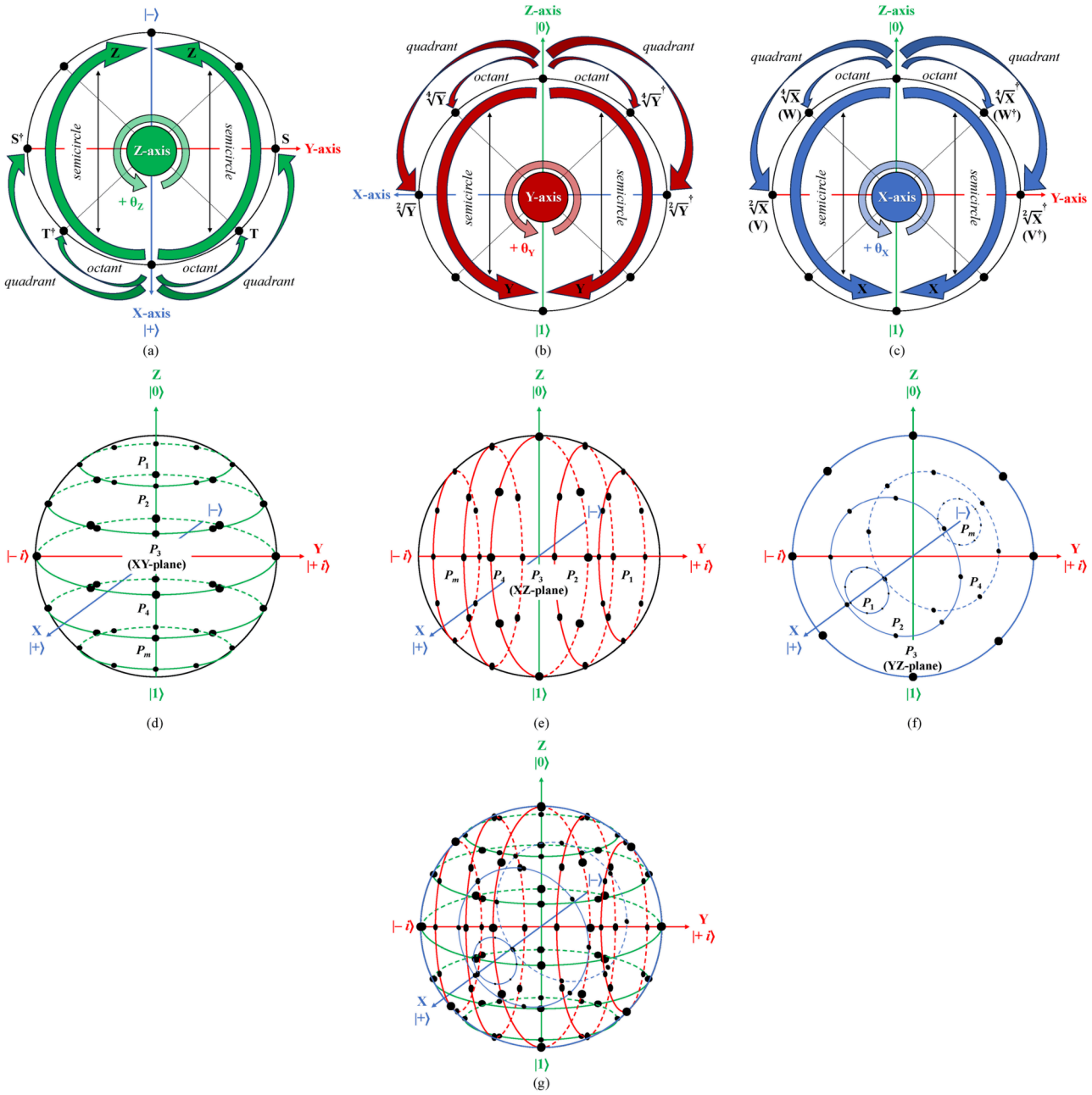

FIGURE 3: Schematics of coordinate and parallel planes (P) perpendicular to various axes intersected with the Bloch sphere.

(a) The XY-plane perpendicular to the Z-axis, as the P_3 in (d), (b) the XZ-plane perpendicular to the Y-axis, as the P_3 in (e), (c) the YZ-plane perpendicular to the X-axis, as the P_3 in (f), (d) m parallel planes perpendicular to the Z-axis, (e) m parallel planes perpendicular to the Y-axis, (f) m perpendicular to the X-axis, and (g) three m parallel planes, each plane perpendicular to one axis of the Bloch sphere, where $m \geq 1$ and the black dots subdividing their axial intersections with the Bloch sphere into the semicircles, quadrants, and octants [22,27,29].

In this paper, we introduce a new methodology using the intersections of different coordinate and parallel planes to geometrically design various cost-effective n-bit quantum gates. Hence, our methodology utilizes the Bloch sphere as a "geometrical design tool" to visually construct $n-bit$ quantum gates. We termed this methodology the "Bloch sphere

approach (BSA)." The cost-effectiveness of the BSA mainly relies on the aforementioned three observations discussed in the Introduction section, utilization of Clifford+T gates, and symmetrical circuit structures.

In previous studies [7,8,21,22,27,29], we discussed how to visually design any *n*-bit quantum gates using the BSA for IBM quantum computers. Because the quantum operations of all IBM native gates mainly rotate the states of a qubit around the X-axis and Z-axis of the Bloch sphere, we utilize the XY-plane of the Bloch sphere (shown in Figure *3*(a)) to visually design different *n*-bit quantum gates. The intersections of the XY-plane with the Bloch sphere are divided into a number of segments to represent the quantum rotations ($\pm\boldsymbol{\theta}$) of IBM native RZ gates applied to a qubit, as stated in three groups below, where $-\boldsymbol{\pi} \leq \boldsymbol{\theta} \leq +\boldsymbol{\pi}$ radians. We choose a quantum base state on each circle of intersection and use the images of that state under various quantum rotations to subdivide the circle of intersection in various ways.

1. The "semicircle" segment is half of the XY-plane's intersection with the Bloch sphere that represents the quantum rotations of Z gates, i.e., we use our base state's images under the RZ($\pm\boldsymbol{\pi}$) gate to subdivide our intersection circle. Here, we then consider our intersection circle to consist of two semicircles.

2. The "quadrant" segments are obtained by replacing the RZ($\pm\boldsymbol{\pi}$) gate in group (1) above, with either of the S and $\mathbf{S}^{\dagger}$ gates, i.e., RZ($+\frac{\boldsymbol{\pi}}{\mathbf{2}}$) and RZ($-\frac{\boldsymbol{\pi}}{\mathbf{2}}$) and their repetition of gates, respectively. Here, our intersection circle consists of four quadrants.

3. The "octant" segments are obtained by replacing the RZ($\pm\boldsymbol{\pi}$) gate in group (1) above, with either of the T and $\mathbf{T}^{\dagger}$ gates, i.e., RZ($+\frac{\boldsymbol{\pi}}{\mathbf{4}}$) and RZ($-\frac{\boldsymbol{\pi}}{\mathbf{4}}$) and their repetition of gates, respectively. Here, our intersection circle consists of eight octants.

Notably, in the XY-plane, the IBM native X, $\sqrt{\mathbf{X}}$, and RX gates rotate the states of a qubit around the X-axis by $+\boldsymbol{\pi}$, $+\frac{\boldsymbol{\pi}}{\mathbf{2}}$, and $\pm\boldsymbol{\theta}$ radians, respectively. However, the IBM native CNOT gate rotates the state of its target qubit around the X-axis by $+\boldsymbol{\pi}$ radians, when its control qubit is set to the state of $|\mathbf{1}\rangle$. Otherwise, no rotations occur.

Generally, the BSA would also be utilized to build generic and cost-effective *n*-bit quantum gates for other quantum computers, e.g., Intel, Google, and Rigetti, using other coordinate and parallel planes with the Bloch sphere, e.g., the XZ-plane and YZ-plane as demonstrated in Figure *3*(b) and (c), respectively, based on the supported Clifford+T gates as native gates for such quantum computers. Please observe that our cross-platform generalized visualization approach is speculative. Selecting another plane of the Bloch sphere may not by itself support a platform's native entanglement, layout connectivity, gate durations, and gate errors. Therefore, we introduce the BSA as a generic and open geometrical framework for prospective quantum computing research to build complex, interesting, and innovative *n*-bit quantum gates, circuits, and libraries.

The BSA design steps

In the BSA, the following steps express how to geometrically design cost-effective *n*-bit quantum gates using the XY-plane, Clifford+T gates (CTG), and symmetrical circuit structures (see Figure *2*) for $\boldsymbol{n-1}$ controls and one target, where $\boldsymbol{n \geq 2}$ qubits.

Step 1: Describe the desirable operation (Boolean logic and quantum behavior) of the to-be-built *n*-bit gate using the truth table [24], Karnaugh map [24], binary decision diagram (BDD) [30,31], just to name a few.

Step 2: Transform the quantum state of the target from the Z-axis of the Bloch sphere into the XY-plane using one H gate. Notice that the target is initially set to either $|\mathbf{0}\rangle$ or $|\mathbf{1}\rangle$ state, depending on the gate's operational purpose. The H gate can be decomposed into a sequence of three Clifford gates {S, $\sqrt{\mathbf{X}}$, S}.

Step 3: For the target, define all Clifford+T gates as the initial set $\mathbf{CTG_0}$ = {H, X, $\sqrt{\mathbf{X}}$, Z, S, $\mathbf{S}^{\dagger}$, T, $\mathbf{T}^{\dagger}$, CNOT\}. The CNOT gate can be generated in the sequence {H, CZ, H}.

Step 4: If the target is not controlling other qubits, then $\mathbf{CTG_0}$ is limited to the set $\mathbf{CTG_1}$ = {H, X, $\sqrt{\mathbf{X}}$, Z, S, $\mathbf{S}^{\dagger}$, T, $\mathbf{T}^{\dagger}$}. Otherwise, $\mathbf{CTG_1} = \mathbf{CTG_0}$.

Step 5: For the target, define all segments of the XY-plane as the initial set $\mathbf{SEG_0}$ = {semicircles, quadrants, octants}.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Step 6: Since $\mathbf{SEG_0}$ only identifies quantum gates rotating around the Z-axis of the Bloch sphere, then $\mathbf{CTG_1}$ is limited to $\mathbf{CTG_2}$ = {Z, S, $\mathbf{S^\dagger}$, T, $\mathbf{T^\dagger}$}. Assume that the quantum rotations around the X-axis of the Bloch sphere are not required for the target. Otherwise, this step is negligible and $\mathbf{CTG_2} = \mathbf{CTG_1}$.

Step 7: Let $n_{\mathbf{CNOT}}$ count the total number of CNOT gates from the controls to the target, and the new sets $\mathbf{CTG_3}$ and $\mathbf{SEG_1}$ will be defined as follows. Notice that the $n_{\mathbf{CNOT}}$ defines the minimum utilized segments (semicircles, quadrants, and/or octants) of the Bloch sphere, as a θ angular rotation divider, which can be generalized to $\theta \leq \pm\frac{\pi}{n_{\mathbf{CNOT}}+1}$.

1. If $n_{\mathbf{CNOT}} = \mathbf{1}$, then $\mathbf{CTG_3}$ = {S, $\mathbf{S^\dagger}$, T, $\mathbf{T^\dagger}$} and $\mathbf{SEG_1}$ = {quadrants, octants}.

2. If $n_{\mathbf{CNOT}} = \mathbf{2}$, then a set of IBM native RZ gates defines $\mathbf{CTG_3}$ = {RZ(θ) | $\theta \leq \pm\frac{\pi}{3}$} for arbitrary $\mathbf{SEG_1}$.

3. If $n_{\mathbf{CNOT}} = \mathbf{3}$, then $\mathbf{CTG_3}$ = {T, $\mathbf{T^\dagger}$} and $\mathbf{SEG_1}$ = {octants}.

4. If $n_{\mathbf{CNOT}} > \mathbf{3}$, then a set of IBM native RZ gates defines $\mathbf{CTG_3}$ = {RZ(θ) | $\theta \leq \pm\frac{\pi}{n_{\mathbf{CNOT}}+1}$} for arbitrary $\mathbf{SEG_1}$.

Step 8: The $n_{\mathbf{CNOT}}$ and the sequence of the permutative gates of $\mathbf{CTG_3}$ should match the described quantum operation (logic and behavior) in Step 1 above, which can be designed and visualized based on the found $\mathbf{SEG_1}$ of the XY-plane.

Step 9: Re-transform the final reached quantum state of the target from the XY-plane into the Z-axis of the Bloch sphere using another H gate.

Step 10: Finally, the re-transformed quantum state of the target is considered the output of such a gate. For instance, the $|\mathbf{1}\rangle$ state is a solution, and the $|\mathbf{0}\rangle$ state is a non-solution.

Weighted transpilation quantum cost and layouts

In previous studies [7,27], we discussed a new metric of the final quantum cost calculation for a transpiled quantum circuit into the limited connectivity layout of any quantum computer, which was termed the "transpilation quantum cost (TQC)". However, in this paper, we introduce a generic variant metric of the TQC, which is termed the "weighted transpilation quantum cost (WTQC)". The WTQC expressed in Eq. (2) is the weighted sum of four components (as listed below) for a final transpiled quantum circuit, where W_i is the weighted utilization importance for each N_1, N_2, XC, and D components and $W_i \geq 0$.

1. N_1 is the total number of all single-qubit native gates in a transpiled circuit.

2. N_2 is the total number of all two-qubit native gates in a transpiled circuit.

3. XC is the total number of all decomposed SWAP gates (see Figure 4), as the crossing connections among the utilized physical neighboring and non-neighboring qubits.

4. D is the depth, as the critical longest path of N_1, N_2, and XC, of a transpiled circuit.

$$\mathbf{WTQC} = W_1 \cdot N_1 + W_2 \cdot N_2 + W_3 \cdot XC + W_4 \cdot D \tag{2}$$

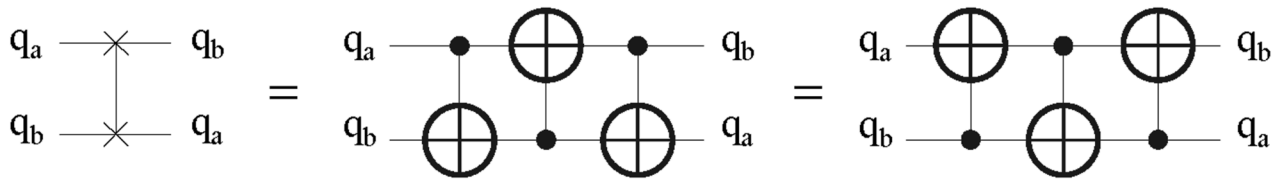

FIGURE 4: Schematics of a SWAP gate (left side) and its equivalent decompositions.

The gate contains three CNOT gates (middle and right sides), for swapping the states of two arbitrary-indexed physical neighboring qubits ($\mathbf{q_a}$ and $\mathbf{q_b}$).

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Notably, for all $W_i = 1$ in Eq. (2), if an *n*-bit quantum gate is designed using the BSA, the WTQC for its final transpiled quantum circuit can then be immediately predicted for a specific QPU, since such a gate was initially constructed using the supported native gates for that QPU. If an *n*-bit quantum gate only utilizes a specific set of physical neighboring qubits, then $XC = 0$ and W_4 is negligible. For instance, Figure *5* illustrates arbitrary sets of n physical neighboring qubits for the heavy-hex layout of ibm_torino QPU of 133 qubits, where (i) $2 \leq n \leq 5$ qubits, (ii) the colors of physical qubits (circles) indicate the characteristics of single-qubit native gates, such as the T1 (relaxation time in μs), T2 (decoherence time in μs), readout errors, etc., (iii) the colors of channels (lines) indicate the characteristics of two-qubit native gates, such as gates length (in ns) and errors, and (iv) the circles and lines are scaled from lower (darker color) to higher (lighter color) values in points (ii) and (iii) above [14,15].

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

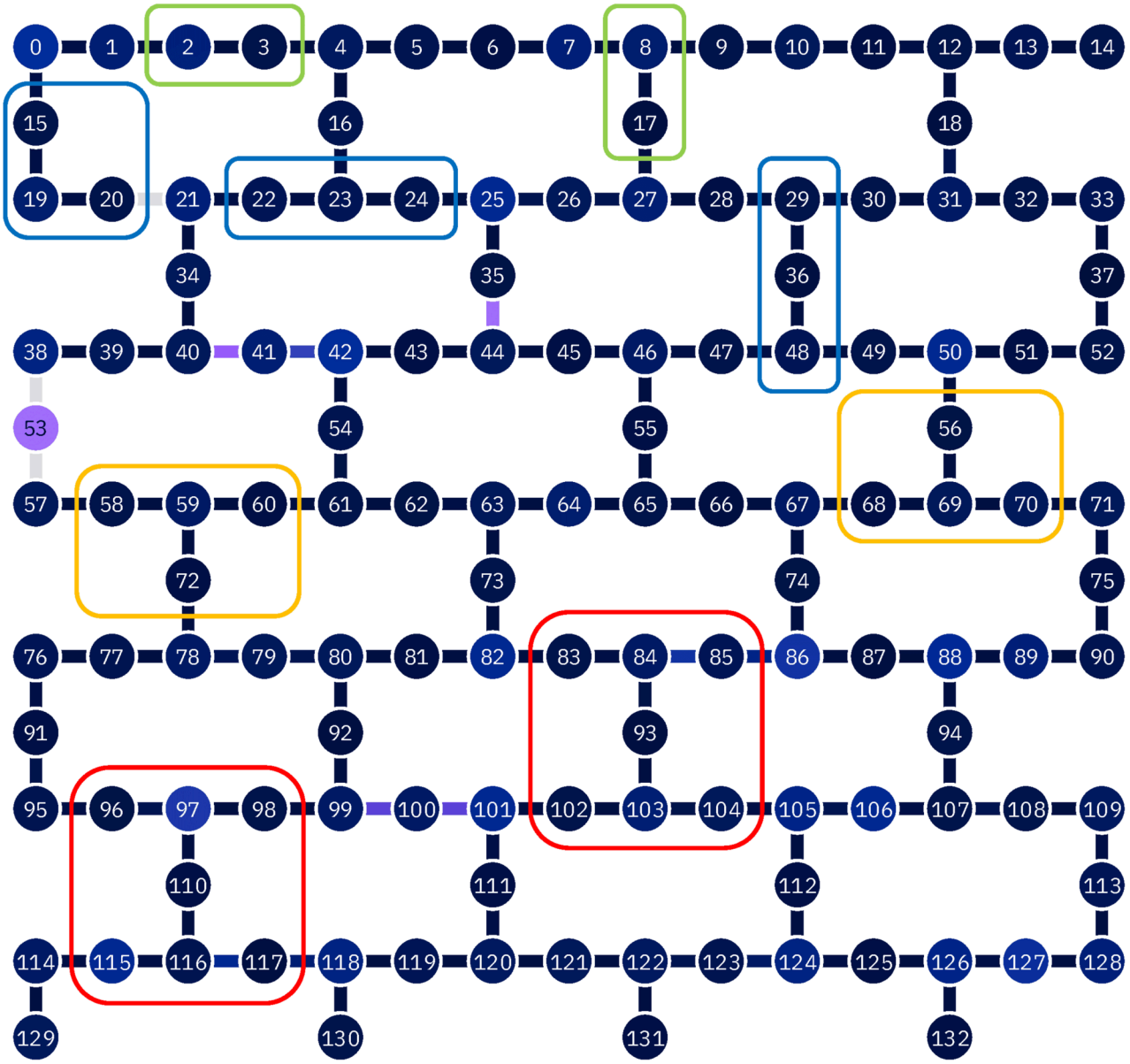

FIGURE 5: The heavy-hex layout of ibm_torino QPU of 133 qubits illustrating arbitrary sets of n physical neighboring qubits.

(i) Sets of two physical neighboring qubits ($n = 2$) as denoted in green, (ii) sets of three physical neighboring qubits ($n = 3$) as denoted in blue, (iii) sets of four physical neighboring qubits [8] ($n = 4$) as denoted in orange, and (iv) sets of seven physical neighboring qubits ($n = 5$ with two additional ancilla qubits as denoted in red [14,15].
QPU, Quantum Processing Unit

Based on Figure *2*(b) and Figure *5*, a symmetrical 3-bit quantum gate (two controls and one target) can be cost-effectively designed using the BSA, by geometrically mapping the two controls ($\mathbf{control_1}$ and $\mathbf{control_2}$) to the physical qubits indexed "15" and "20," respectively, and the target between these two controls to the physical qubit indexed "19," as denoted in the left-most blue group. Hence, the WTQC of this 3-bit gate has significantly decreased since $XC = 0$, i.e., SWAP gates are never added to connect these three physical neighboring qubits, with fewer generated *D*.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

For the heavy-hex layout of any IBM QPU and $3 < n \leq 5$ qubits, symmetrical *n*-bit quantum gates can also be cost-effectively designed using the BSA, by selecting different sets of $n-1$ physical neighboring qubits for the controls and one physical neighboring qubit for the target that is mapped among these controls.

Results And Discussion

In our GALA-n [7] and CALA-n [8] quantum libraries, we discussed how to geometrically design cost-effective *n*-bit quantum gates and operators using the BSA, where $2 \leq n \leq 5$ qubits. For brevity and ease of demonstration, we present the aforementioned 10 BSA design steps for geometrically constructing the cost-effective 3-bit Toffoli gate as follows.

Step 1: Describe the Boolean logic (as the quantum behavior) for the to-be-built 3-bit Toffoli gate using the truth table, as stated in Table *2*. Such a described gate only outputs the state of $|1\rangle$ when both controls are in the state of $|1\rangle$. Notice that because this gate only utilizes three qubits without any ancilla qubits, we can immediately design its quantum circuit shown in Figure *2*(b), where its target is initially set to the state of $|0\rangle$.

$\|control_2\rangle$	$\|control_1\rangle$	$\|target\rangle$	Output in Boolean logic
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	False
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	False
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	False
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	True

TABLE 2: The quantum-based truth table for the Boolean logic of a 3-bit Toffoli gate of two control (input) qubits and one target (output) qubit, when the target = $|0\rangle$.

Step 2: Transform the state of the target from the Z-axis of the Bloch sphere into the XY-plane using the H gate, as $\mathbf{SP_1}$ shown in Figure *2*(b).

Step 3: $\mathbf{CTG_0}$ = {H, X, $\sqrt{\mathbf{X}}$, Z, S, $\mathbf{S^\dagger}$, T, $\mathbf{T^\dagger}$, CNOT} for all θ's gates in Figure *2*(b).

Step 4: Because the target is not controlling other qubits, $\mathbf{CTG_1}$ = {H, X, $\sqrt{\mathbf{X}}$, Z, S, $\mathbf{S^\dagger}$, T, $\mathbf{T^\dagger}$} for all θ's gates shown in Figure *2*(b).

Step 5: For the target, $\mathbf{SEG_0}$ = {semicircles, quadrants, octants}, as shown in Figure *3*(a).

Step 6: For the target, $\mathbf{CTG_2}$ = {Z, S, $\mathbf{S^\dagger}$, T, $\mathbf{T^\dagger}$}, because there are no quantum rotation requirements by the X-axis of the Bloch sphere, for all θ's gates shown in Figure *2*(b).

Step 7: $n_{\mathrm{CNOT}} = 3$, $\mathbf{CTG_3}$ = {T, $\mathbf{T^\dagger}$} for all θ's gates in Figure *2*(b) and $\mathbf{SEG_1}$ = {octants}.

Step 8: The n_{CNOT} and the sequence of the permutative gates of $\mathbf{CTG_3}$ should geometrically match the quantum-based Boolean logic expressed in Table *2*, and Figure *6* geometrically visualizes this step.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

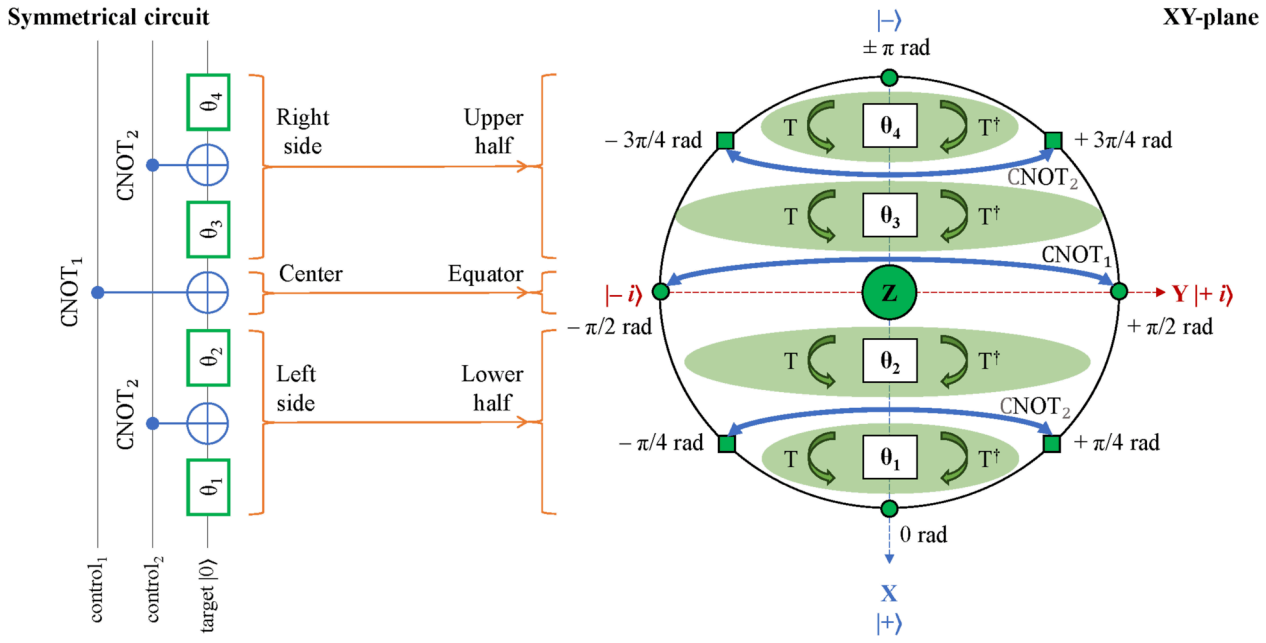

FIGURE 6: The geometrical matching visualization between the quantum gates of the symmetrical circuit of the 3-bit Toffoli gate and their quantum rotations in the XY-plane.

where all quantum operations (gates and rotations) by the X-axis and the Z-axis are denoted in blue and green, respectively, and each θ has a set of permutative gates $\mathbf{CTG_3}$ = {T, $\mathbf{T^\dagger}$}. Notice that $\mathbf{SP_1}$ and $\mathbf{SP_2}$ (as H) gates are omitted in this figure, since the target is now in the XY-plane [7,8,27].

Step 9: Based on Table *2* and Figure *6*, the final reached state of the target is re-transformed from the XY-plane into the Z-axis of the Bloch sphere using the H gate, as the $\mathbf{SP_2}$ shown in Figure *2*(b).

Step 10: The re-transformed quantum state of the target is the output of this 3-bit Toffoli gate, i.e., the target in the state of $|1\rangle$ indicates a solution when both controls are in the state of $|1\rangle$. Otherwise, as a non-solution ($|0\rangle$) for all other combinations of both controls, as expressed in Table *2*.

Please observe that, in **Step 8** above, we found that the accurate permutative gates of $\mathbf{CTG_3}$ for all θ's are in the sequence {θ_1 = $\mathbf{T^\dagger}$, θ_2 = T, θ_3 = $\mathbf{T^\dagger}$, θ_4 = T\}, to successfully design this 3-bit Toffoli gate matching its quantum-based Boolean logic in Table *2*. Analytically, Table *3* verifies such accurate gate selections for $\mathbf{CTG_3}$ for all θ's, where $\frac{\alpha}{\beta}\frac{\pi}{\pi} = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\frac{\alpha}{\beta}\frac{\pi}{\pi}}|1\rangle\right)$, α and $\beta \in \mathbb{R}$, $e^{\pm ix} = \cos(x) \pm i\sin(x)$, and "-" means a gate is not applied to the target [7,8,27]. Please observe that, in Table *3*, all local phases generated by the target (output) qubit of this 3-bit Toffoli gate are ignored, since the Boolean-based outputs (as states and not phases) are only counted for the final Boolean-based decisions.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

|control 2 control$_1$ ⟩	SP$_1$	θ$_1$	CNOT$_2$	θ$_2$	CNOT$_1$	θ$_3$	CNOT$_2$	θ$_4$	SP$_2$	Boolean output
	(H)	($T^{\dagger}$)		(T)		($T^{\dagger}$)		(T)	(H)	
|0 0⟩	|+⟩	7π / 4	-	|+⟩	-	7π / 4	-	|+⟩	|0⟩	False
|0 1⟩	|+⟩	7π / 4	-	|+⟩	|+⟩	7π / 4	-	|+⟩	|0⟩	False
|1 0⟩	|+⟩	7π / 4	π / 4	|+ i⟩	-	π / 4	7π / 4	|+⟩	|0⟩	False
|1 1⟩	|+⟩	7π / 4	π / 4	|+ i⟩	|- i⟩	5π / 4	3π / 4	|-⟩	|1⟩	True

TABLE 3: Analytical verification for the accurate gate selections for all θ's of the 3-bit Toffoli gate.

Source: [7,8,27]

In our research [8,27], we also investigated other permutation orders and sequences for the gate selection of the found $\mathbf{CTG_3}$ in Step 8 above, by adding two auxiliary (AX) quantum gates to construct various 3-bit quantum Boolean-based gates as shown in Figure *7*. Subsequently, the constructed 3-bit quantum Boolean-based operators are the AND, NAND, OR, NOR, implication, and inhibition gates. Table *4* analytically states these permutation orders and sequences of $\mathbf{CTG_3}$ for all $\boldsymbol{\theta}$'s of a 3-bit symmetrical circuit shown in Figure *7*, where the target is initially set to the state of $|\mathbf{0}\rangle$. Notice that, in Figure *7*, the $\mathbf{AX_1}$ gate is reserved for future work and investigations to construct other useful quantum Boolean-based and phase-based operators.

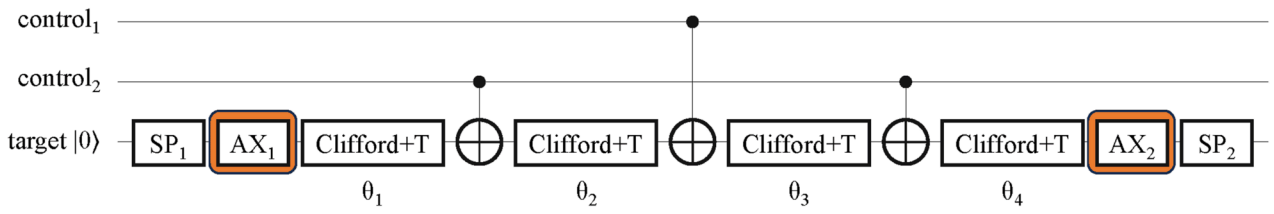

FIGURE 7: Our generalized symmetrical circuit structure for the 3-bit quantum gate of CALA-n.

where all $\boldsymbol{\theta}$'s (as $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$, $\boldsymbol{\theta_3}$, and $\boldsymbol{\theta_4}$) are Clifford+T gates, $\mathbf{SP_1}$ and $\mathbf{SP_2}$ are the superposition gates, $\mathbf{AX_1}$ and $\mathbf{AX_2}$ are the auxiliary gates for constructing various 3-bit quantum Boolean-based gates, and the target qubit is initially set to the state of $|\mathbf{0}\rangle$ [8,27]. CALA-n, Clifford+T-Based Architecture of Layout-Aware n-Bit

3-bit Boolean operators	SP_1	AX_1	θ_1	θ_2	θ_3	θ_4	AX_2	SP_2
AND: (a ∧ b)	H	I	$T^\dagger$	T	$T^\dagger$	T	I	H
NAND: ¬ (a ∧ b)	H	I	$T^\dagger$	T	$T^\dagger$	T	- Z	H
OR: (a ∨ b)	H	I	T	T	T	T	Z	H
NOR: ¬ (a ∨ b)	H	I	T	T	T	T	I	H
Implication: (¬ a ∨ b)	H	I	$T^\dagger$	$T^\dagger$	T	T	- Z	H
Inhibition: ¬ (¬ a ∨ b)	H	I	$T^\dagger$	$T^\dagger$	T	T	I	H

TABLE 4: The gate selections for constructing various 3-bit quantum Boolean-based operators using the BSA.

Source: [8,27]

Notably, in Table *4*, the quantum Boolean-based operators can be effectively used to build Boolean oracles [32-34] and Boolean-Hamiltonians transformed oracles [35,36], especially for the functions and problems that are logically structured in the Product-of-Sums [24,37], Sum-of-Products [24,38], Exclusive-or Sum-of-Products [39,40], constraints satisfaction problems - satisfiability [41], and in designing XOR functions of two product gates of different Hamming distances [42], just to name a few. Please observe that Grover's algorithm is capable of searching for all solutions for a Boolean oracle built from phase-relative Boolean gates when only utilizing our introduced Grover controlled-diffusion operator (CU_s) [33,34], rather than utilizing the standard Grover diffusion operator (U_s) [3,4,32,33], since our (CU_s) only relies on the Boolean-decision output from a Boolean oracle as well as does not rely on the conventional phase kickback technique [3,4,33].

In this paper, the BSA along with the parametric selection and modification of the rotational angles (θ's), superposition gates ($\mathbf{SP_1}$ and $\mathbf{SP_2}$), and auxiliary gates ($\mathbf{AX_1}$ and $\mathbf{AX_2}$) are open topics for future quantum computing research, as an introductory geometrical design framework to construct interesting and cost-effective *n*-bit quantum Boolean-based and phase-based operators relying on the layouts and the number of n physical neighboring qubits for different quantum computers.

Based on GALA-n ($2 \leq n \leq 4$) and CALA-n ($2 \leq n \leq 5$) quantum libraries, various experiments of *n*-bit quantum gates and operators were designed using conventional design approaches and the BSA. These experiments were then transpiled using the ibm_brisbane QPU of 127 qubits [43]. The experiments built using conventional design approaches were transpiled using the standard IBM Transpiler [44], i.e., the logical qubits of these experiments are automatically mapped to their corresponding physical qubits by this Transpiler, while the experiments built using the BSA were transpiled using our manual assignments for the logical qubits to their corresponding physical qubits based on the

layout of this QPU. Notice that, for the transpilation process on the same QPU, one of the key points of using the BSA versus the standard IBM Transpiler is the initial logical-to-physical qubits assignment. Hence, a developer should always use the manual assignment when utilizing the BSA; however, a developer may use either the automatic or manual assignment when utilizing the standard IBM Transpiler. Therefore, for the BSA, a developer should have prior knowledge of the architecture of the restricted layout connectivity for the utilized superconducting quantum computer, for efficient qubit mapping and lower-cost circuit synthesis. In this research, we utilized the following compiler configurations: the identical native (basis) gates, Qiskit version = 2.1, qiskit_ibm_runtime version = 0.42 (including SamplerV2 library), dynamical_decoupling.enable = True (for SamplerV2), dynamical_decoupling.sequence_type = "XX" (for SamplerV2), shots = 1024, optimization_level = 3, and seed_transpiler has the same randomly generated value for both the BSA and the standard IBM Transpiler.

Notice that the Transpiler mostly generates cost-expensive and non-symmetrical structures for transpiled circuits. For instance, Figure *8* illustrates the decomposition of the IBM non-native 3-bit Toffoli gate into a non-symmetrical circuit consisting of IBM native gates as part of IBM transpilation processes. For this reason, SWAP gates are normally added to connect the physical non-neighboring qubits (even for physical neighboring qubits) shown in Figure *5*, thereby increasing the final quantum cost for such transpiled circuits. In contrast, we initially construct circuits using the symmetrical structures (see Figure *2*) to be efficiently utilized by the BSA and to cost-effectively fit the layout geometry of a specific quantum computer. Additionally, our manual physical qubit assignment guarantees to never add SWAP gates. For this reason, the quantum costs of our transpiled circuits always have lower values than those generated by the Transpiler.

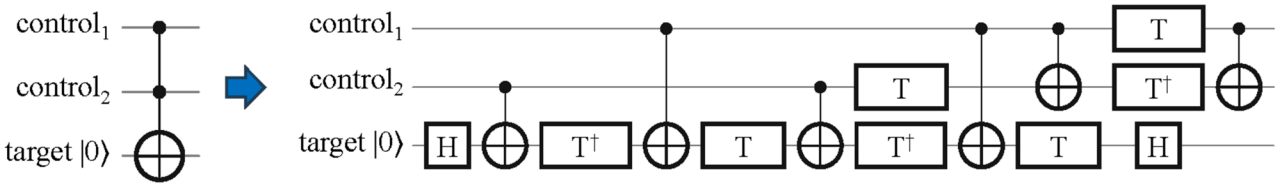

FIGURE 8: The decomposition of an IBM non-native 3-bit Toffoli gate (left side) into a non-symmetrical circuit consisting of IBM native gates (right side).

Notice that SWAP gates are required here to connect the controls and target (as the physical neighboring and non-neighboring qubits) to fit the heavy-hex layout of an IBM QPU.
QPU, Quantum Processing Unit

Table *5* summarizes the cost-effectiveness design approaches of the final transpiled circuits using our proposed WTQC metric, as expressed in Eq. (2), for the experiments generated from: (i) the conventional design approaches, and (ii) the BSA for symmetrical circuit structures.

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Quantum operators	n qubits	Using conventional design approaches					Using our BSA				
		(N_1	N_2	XC	D	WTQC)	(N_1	N_2	XC	D	WTQC)
Controlled-√X	2	7	2	0	9	18	6	1	0	7	14
Controlled-√X	4	147	29	5	110	291	53	7	0	42	102
AND	3	48	5	2	42	97	34	3	0	29	66
AND	4	146	29	5	108	288	52	7	0	41	100
AND	5	387	41	24	299	751	81	9	0	74	164
NAND	4	105	20	2	83	210	52	7	0	41	100
OR	4	102	20	2	80	204	52	7	0	41	100
OR	5	398	41	27	304	770	92	9	0	76	177
NOR	4	103	20	2	83	208	52	7	0	41	100
Implication	3	49	9	1	39	98	28	3	0	21	52
Inhibition	3	50	9	1	39	99	28	3	0	21	52
Fredkin	3	67	7	4	59	137	35	5	0	22	62
Fredkin	4	189	40	8	115	352	56	9	0	46	111
Miller	4	151	29	3	113	296	70	13	0	58	141

TABLE 5: The WTQC of different transpiled circuits of n-bit quantum gates, where all weighted importance $W_i = 1$ and $2 \leq n \leq 5$ physical neighboring qubits for the layout of ibm_brisbane QPU.

BSA, Bloch Sphere Approach; QPU, Quantum Processing Unit; WTQC, Weighted Transpilation Quantum Cost

Accordingly, our proposed BSA as the geometrical design tool successfully generates various cost-effective *n*-bit quantum gates using the symmetrical circuit structure, Clifford+T gates, as well as the supported native gates and layout geometry for a specific QPU. The BSA can provide a broad range of various cost-effective quantum Boolean-based gates

and operators for practical applications in the fields of Boolean oracular and heuristic search algorithms, digital logic circuits, and machine learning, just to name a few.

In our future research, we will fundamentally utilize the BSA to geometrically build cost-effective complex *n*-bit quantum operators, such as arithmetic adders, comparators, and symmetrical logical circuits [45]. The WTQC can be formally used for the evaluation and benchmarking purposes of various cost-effective design approaches using different layouts of quantum computers. Based on our previous research, more design methods should be applied to other types of basic gates, such as the majority gate [46,47], Peres and inverse Peres gates [11,48,49], and EXOR link (Hamming distance "HD") gates [50], to develop and complete useful quantum libraries, especially for arithmetic circuits.

Conclusions

This paper introduces a new quantum circuit design approach using the Bloch sphere as a "geometrical design tool" to build cost-effective *n*-bit quantum gates with lower quantum costs, where $n \geq 2$. This new design is termed the BSA. In the BSA, cost-effective *n*-bit quantum gates are built without using any multiplication of unitary matrices, as this is a fundamental technique used in many conventional design approaches that may require extensive computational time and generate cost-expensive *n*-bit quantum gates. The BSA visually builds an *n*-bit quantum gate using the geometrical planar intersections (as the XY-plane) of the Bloch sphere, as the desirable quantum rotations for a to-be-built gate. To cost-effectively design *n*-bit quantum gates for $n-1$ control qubits and one target qubit, the BSA mainly utilizes these four factors: (i) the XY-plane, (ii) Clifford+T gates, (iii) symmetrical circuit structures, and (iv) efficiently mapping the target qubit among the control qubits to geometrically satisfy the limited layout connectivity of n physical neighboring qubits for a specific quantum computer. We also introduce a new technology-dependent metric to calculate the quantum cost of the final transpiled *n*-bit quantum gates into a specific quantum computer. This new metric is termed the WTQC. The WTQC is the weighted sum of: (i) the total number of single-qubit native gates, (ii) the total number of two-qubit native gates, (iii) the total number of SWAP gates, and (iv) the depth of a transpiled quantum circuit. Experimentally, using an IBM quantum computer, various *n*-bit quantum gates designed using the BSA always have lower WTQC values than those for such gates designed using conventional design techniques, for $2 \leq n \leq 5$ qubits.

In conclusion, on the one hand, the BSA as a geometrical design tool successfully creates different cost-effective *n*-bit quantum gates for IBM quantum computers, with the practical approval of using our introduced WTQC as a technology-dependent cost metric. On the other hand, the BSA along with different utilization of other geometrical planar intersections (as the XZ-plane and YZ-plane) of the Bloch sphere and other quantum rotational gates (including Clifford+T gates) is an open future quantum computing research in the topics of constructing many interesting and cost-effective *n*-bit quantum Boolean-based and phase-based operators for Boolean and phase oracles, respectively, which are geometrically restricted to the limited layouts and the number of *n* physical neighboring qubits for different quantum computers.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Ali Al-Bayaty

Acquisition, analysis, or interpretation of data: Ali Al-Bayaty, Shanyan Chen, Ali Al-Shuwaili, Abbas AlZubaidi, Marek Perkowski

Drafting of the manuscript: Ali Al-Bayaty, Shanyan Chen, Ali Al-Shuwaili, Abbas AlZubaidi, Marek Perkowski

Critical review of the manuscript for important intellectual content: Ali Al-Bayaty, Shanyan Chen, Ali Al-Shuwaili, Abbas AlZubaidi, Marek Perkowski

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

Supervision: Ali Al-Bayaty, Marek Perkowski

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

All data generated or analyzed during this study are included in this published article and/or its appendices.

References

1. Barenco A, Bennett CH, Cleve R, et al.: Elementary gates for quantum computation. Physical Review A. 1995, 52:3457. 10.1103/physreva.52.3457
2. Ralph T, Resch K, Gilchrist A: Efficient Toffoli gates using qudits. Physical Review A-Atomic, Molecular, and Optical Physics. 2007, 75:022313. 10.1103/physreva.75.022313
3. Nielsen MA, Chuang IL: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge; 2010.
4. LaPierre R: Introduction to Quantum Computing. Springer, Cham, Switzerland; 2021. 10.1007/978-3-030-69318-3
5. Smolin JA, DiVincenzo DP: Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. Physical Review A. 1996, 53:2855-2856. 10.1103/physreva.53.2855
6. Patel RB, Ho J, Ferreyrol F, Ralph TC, Pryde GJ: A quantum Fredkin gate. Science Advances. 2016, 2:e1501531. 10.1126/sciadv.1501531
7. Al-Bayaty A, Perkowski M: GALA-n: Generic architecture of layout-aware n-bit quantum operators for cost-effective realization on IBM quantum computers [PREPRINT]. arXiv. 2023, 10.48550/arXiv.2311.06760
8. Al-Bayaty A, Song X, Perkowski M: CALA-n: A quantum library for realizing cost-effective 2-, 3-, 4-, and 5-bit gates on IBM quantum computers using Bloch sphere approach, Clifford+T gates, and layouts [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2408.01025
9. Lee S, Lee SJ, Kim T, Lee JS, Biamonte J, Perkowski M: The cost of quantum gate primitives. Journal of Multiple-Valued Logic & Soft Computing. 2006, 12:561.
10. Hung WN, Song X, Yang G, Yang J, Perkowski M: Optimal synthesis of multiple output Boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2006, 25:1652-1663. 10.1109/tcad.2005.858352
11. Szyprowski M, Kerntopf P: Low quantum cost realization of generalized Peres and Toffoli gates with multiple-control signals. 2013 13th IEEE International Conference on Nanotechnology (IEEE-NANO 2013). 2013, 802-807. 10.1109/NANO.2013.6721034
12. Shende VV, Markov IL: On the CNOT-cost of TOFFOLI gates. Quantum Information & Computation. 2009, 9:461-486.
13. Schmitt B and De Micheli G: Tweedledum: A compiler companion for quantum computing. 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). 2022, 7-12. 10.23919/DATE54114.2022.9774510

14. Baglio J: Cross-platform hardware benchmark of style-based quantum GANs for data augmentation on superconducting and trapped-ion processors. arXiv. 2024, 10.48550/arXiv.2405.04401

15. Belaloui NE, Tounsi A, Khamadja AR, et al.: Ground-state energy estimation on current quantum hardware through the variational quantum eigensolver: A practical study. Journal of Chemical Theory and Computation. 2025, 21:6777-6792. 10.1021/acs.jctc.4c01657

16. Wille R, Lye A, Drechsler R: Optimal SWAP gate insertion for nearest neighbor quantum circuits. 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC). 2014, 489-494. 10.1109/ASPDAC.2014.6742939

17. Gokhale P, Tomesh T, Suchara M, Chong F: Faster and more reliable quantum swaps via native gates. Proceedings of the 2024 International Conference on Parallel Architectures and Compilation Techniques. 2024, 351-362. 10.1145/3656019.3689818

18. Mi A, Deng S, Szefer J: Securing reset operations in NISQ quantum computers. Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. 2022, 2279-2293. 10.1145/3548606.3559380

19. Murali P, Baker JM, Javadi-Abhari A, Chong FT, Martonosi M: Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems. 2019, 1015-1029. 10.1145/3297858.3304075

20. Koch D, Martin B, Patel S, Wessing L, Alsing PM: Demonstrating NISQ era challenges in algorithm design on IBM's 20 qubit quantum computer. AIP Advances. 2020, 10:1-14. 10.1063/5.0015526

21. Al-Bayaty A, Perkowski M: Cost-effective realization of n-bit Toffoli gates for IBM quantum computers using the Bloch sphere approach and IBM native gates [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2410.13104

22. Al-Bayaty A, Chen S, Bleiler SA, Perkowski M: A cost-effective quantum Boolean-Phase SWAP gate with only two CNOT gates [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2507.17164

23. Bravyi S, Kitaev A: Universal quantum computation with ideal Clifford gates and noisy ancillas. Physical Review A. 2005, 71:022316. 10.1103/physreva.71.022316

24. Wakerly JF: Digital Design Principles and Practices. Pearson Education, New York, NY, USA; 2018.

25. IBM Quantum Computing | Qiskit Ecosystem. Accessed: December 15, 2025: https://www.ibm.com/quantum/ecosystem.

26. Maslov D, Dueck GW: Improved quantum cost for n-bit Toffoli gates. Electronics Letters. 2003, 39:1790-1791. 10.1049/el:20031202

27. Al-Bayaty A: Layout-Aware Quantum Circuitry and Algorithmic Extensions to Grover's Algorithm. Portland State University, Portland, OR, USA; 2025. 10.15760/etd.3924

28. Al-Bayaty A, Al-Shuwaili A, AlZubaidi A, Perkowski M: Improving the circuit realization of Grover's quantum search algorithm by replacing Hadamard with √X gates. Proceedings of the 2026 2nd International Conference on Computing and Emerging Sciences (ICCES '26). 2026, 258-265. 10.1145/3797491.3797543

29. Al-Bayaty A, Perkowski M: A geometrical design tool for building cost-effective layout-aware n-bit quantum gates using the Bloch sphere approach [PREPRINT]. arXiv. 2026, 10.48550/arXiv.2601.00484

30. Ebendt R, Fey G, Drechsler R: Advanced BDD Optimization. Springer, New York, NY; 2005. 10.1007/b107399

31. Wille R, Drechsler R: Effect of BDD optimization on synthesis of reversible and quantum logic. Electronic Notes in Theoretical Computer Science. 2010, 253:57-70. 10.1016/j.entcs.2010.02.006

32. Figgatt C, Maslov D, Landsman KA, Linke NM, Debnath S, Monroe C: Complete 3-qubit Grover search on a programmable quantum computer. Nature Communications. 2017, 8:1918. 10.1038/s41467-017-01904-7

33. Al-Bayaty A, Perkowski M: A concept of controlling Grover diffusion operator: A new approach to solve arbitrary Boolean-based problems. Scientific Reports. 2024, 14:23570. 10.1038/s41598-024-74587-y

34. Al-Bayaty A, Perkowski M: Grover controlled-diffuser (CUs) for quantum Boolean oracles of Grover's algorithm. protocols.io. 2024, 10.17504/protocols.io.e6nvw1wj9lmk/v2

35. Al-Bayaty A, Perkowski M: BHT-QAOA: The generalization of quantum approximate optimization algorithm to solve arbitrary Boolean problems as Hamiltonians. Entropy. 2024, 26:843. 10.3390/e26100843

36. Al-Bayaty A, Perkowski M: The impact of optimization approximation algorithms on the performance of the BHT-QAOA. Academia Quantum. 2025, 2:1-12. 10.20935/acadquant7996

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5

37. Astola JT, Stanković RS: Fundamentals of Switching Theory and Logic Design: A Hands on Approach. Springer, New York, NY; 2006. 10.1007/0-387-30311-1

38. Zimmermann R, Tran DQ: Optimized synthesis of sum-of-products. The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers. 2003, 1:867-872. 10.1109/ACSSC.2003.1292036

39. Mishchenko A, Perkowski M: Fast heuristic minimization of exclusive-sums-of-products. Proceedings of RM'2001 Workshop. 2001, 242-250.

40. Sasao T: EXMIN2: A simplification algorithm for exclusive-OR-sum-of-products expressions for multiple-valued-input two-valued-output functions. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1993, 12:621-632. 10.1109/43.277608

41. Perkowski M: Inverse problems, constraint satisfaction, reversible logic, invertible logic and Grover quantum oracles for practical problems. Science of Computer Programming. 2022, 218:102775. 10.1016/j.scico.2022.102775

42. Schaeffer B, Tran L, Gronquist A, Perkowski M, Kerntopf P: Synthesis of reversible circuits based on products of exclusive or sums. 2013 IEEE 43rd International Symposium on Multiple-Valued Logic. 2013, 35-40. 10.1109/ISMVL.2013.54

43. Farrell RC, Illa M, Ciavarella AN, Savage MJ: Scalable circuits for preparing ground states on digital quantum computers: The Schwinger model vacuum on 100 qubits. PRX Quantum. 2024, 5:020315. 10.1103/prxquantum.5.020315

44. Wilson E, Singh S, Mueller F: Just-in-time quantum circuit transpilation reduces noise. 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). 2020, 345-355. 10.1109/QCE49297.2020.00050

45. Chen S, Al-Bayaty A, Song X, Perkowski M: Stesso: A reconfigurable decomposition of n-bit Toffoli gates using symmetrical logical structures and adjustable support qubits [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2510.26116

46. Yang G, Hung WN, Song X, Perkowski M: Majority-based reversible logic gates. Theoretical Computer Science. 2005, 334:259-274. 10.1016/j.tcs.2004.12.026

47. Seyedi S, Jafari Navimipour N: Designing a multi-layer full-adder using a new three-input majority gate based on quantum computing. Concurrency and Computation: Practice and Experience. 2022, 34:e6653. 10.1002/cpe.6653

48. Tsai E: Methodologies for Quantum Circuit and Algorithm Design at Low and High Levels. Portland State University, Portland, OR; 2022. 10.15760/etd.7927

49. Donald J, Jha NK: Reversible logic synthesis with Fredkin and Peres gates. ACM Journal on Emerging Technologies in Computing Systems. 2008, 4:1-19. 10.1145/1330521.1330523

50. Song N, Perkowski MA: Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 1996, 15:385-395. 10.1109/43.494702

How to cite this article:
Al-Bayaty A, Chen S, Al-Shuwaili A, et al. (September 15, 2026) A Geometrical Design Tool for Building Cost-Effective Layout-Aware n-Bit Quantum Gates Using the Bloch Sphere Approach. Cureus J Comput Sci 3 : es44389-026-00279-5. DOI https://doi.org/10.7759/s44389-026-00279-5